Tight Approximation Bounds for Connectivity with a Color-Spanning Set*

Chenglin Fan[1], Jun Luo[1,2], and Binhai Zhu[3]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
{cl.fan,jun.luo}@siat.ac.cn
[2] Huawei Noah's Ark Laboratory, Hong Kong
[3] Department of Computer Science, Montana State University, Bozeman, MT 59717, USA
bhz@cs.montana.edu

Abstract. Given a set of points Q in the plane, define the $\frac{r}{2}$-*Disk Graph*, $Q(r)$, as a generalized version of the Unit Disk Graph: the vertices of the graph is Q and there is an edge between two points in Q iff the distance between them is at most r. In this paper, motivated by applications in wireless sensor networks, we study the following geometric problem of color-spanning sets: given n points with m colors in the plane, choosing m points P with distinct colors such that the $\frac{r}{2}$-Disk Graph, $P(r)$, is connected and r is minimized. When at most two points are of the same color c_i (or, equivalently, when a color c_i spans at most two points), we prove that the problem is NP-hard to approximate within a factor $3 - \varepsilon$. And we present a tight factor-3 approximation for this problem. For the more general case when each color spans at most k points, we present a factor-($2k$-1) approximation. Our solutions are based on the applications of the famous Hall's Marriage Theorem on bipartite graphs, which could be useful for other problems.

1 Introduction

In a wireless sensor network (WSN), the typical objective is to use a set of sensors (modelled as unit disks) to cover a region (or a set of objects) completely. However, in many situations this is either impossible or too costly to achieve, like in a battlefield or in a vast rural area. Hence, recently *partial covers* are proposed to cover a region (or a set of objects) with a decent quality guarantee [18, 27]. (It is well-known that in WSNs the communication range is greater than the sensing range and if the former is at least twice the latter then a complete coverage implies a communication connectivity [26].) Certainly, in partial covers the sensors are usually disconnected (within their sensing range), so we need to increase the communication range (radius) to make the whole WSN connected — which certainly takes energy.

In Figure 1, we show a partial cover with three connected components/clusters A, B and C. To save energy, we just need to select three leaders a, b, c respectively so that by increasing the communication range of these leaders they can communicate with each

* This research is partially funded by the International Science and Technology Cooperation Program of China (2010DFA92720-08-1) and NSF of China under project 11271351.

L. Cai, S.-W. Cheng, and T.-W. Lam (Eds.): ISAAC2013, LNCS 8283, pp. 590–600, 2013.

other and possibly relay some important sensing data from each cluster. In fact, in a more general and slightly different setting, say, in a social network where the members in each group can communicate in different ways, the clusters could even be interleaved and might be inseparable geometrically.

This is often referred to as the color-spanning problem in computational geometry, usually to handle imprecise data. In this model, each imprecise point is modelled as a set T of discrete points which can all be painted by one distinct color c_i (we also say that the color c_i *spans* T). The causes of imprecise data can be various, for example, the uncertain properties of a moving object [5], measurement error, sampling error, network latency [20,21], location privacy protection [3,6,12], etc. Any or a combination of these factors could be leading to imprecision of the data, hence such a new model makes sense for many applications.

In the database area, a similar framework under a different name "uncertain data" has also been used. An imprecise point is called an uncertain object and the different positions with the same color are regarded as the different possible instances of an uncertain object. Pei *et al.* have performed some research that pertains to geometric problems in this framework [4, 19, 24].

In general, the color-spanning problem is to select exactly one point from each colored point set such that certain properties (e.g. area, distance, perimeter, etc) of some underlying geometric structures (e.g. convex hulls, minimum spanning trees, etc) based on the selected points with different colors are minimized or maximized. We give a brief review for some works in computational geometry below.

In the following review, we assume that there are n points with m colors for the sake of notation consistency. Zhang *et al.* [25] proposed a brute force algorithm to address the minimum diameter color-spanning set problem (MDCS). The running time is $O(n^m)$. Fleischer and Xu [11] showed that the MDCS problem can be solved in polynomial time for the L_1 and L_∞ metrics, while it is NP-hard for all other L_p metrics (even for $p = 2$). They also gave an efficient algorithm to compute a constant factor approximation.

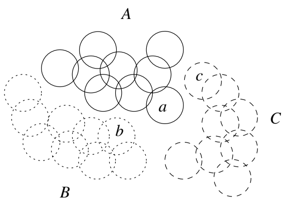

Fig. 1. Three connected components A, B and C for a partial cover. a, b and c can be selected to use the minimum energy to make them connected via communication.

Abellanas *et al.* [1] showed that the Farthest Color Voronoi Diagram (FCVD) is of complexity $\Theta(nm)$ if $m \leq n/2$. Then they proposed algorithms to construct FCVD, the smallest color-spanning circle based on FCVD, the smallest color-spanning rectangle and the narrowest color-spanning strip of arbitrary orientation. In [2], Abellanas *et al.* also proposed an $O(\min\{n(n-m)^2, nm(n-m)\})$ time algorithm for computing the smallest perimeter axis-parallel rectangle enclosing at least one point of each color.

In [9], Das *et al.* proposed an algorithm for identifying the smallest color-spanning corridor in $O(n^2 \log n)$ time and $O(n)$ space and an algorithm for identifying the smallest color-spanning rectangle of arbitrary orientation with an $O(n^3 \log m)$ running time and $O(n)$ space.

Ju *et al.* [16] recently studied several other color-spanning problems. They gave an efficient randomized algorithm to compute a maximum diameter color-spanning set, and they showed it is NP-hard to compute a largest closest pair color-spanning set and a planar minimum color-spanning tree.

Given a set of points Q in the plane, define the $\frac{r}{2}$-*Disk Graph*, $Q(r)$, as a generalized version of the Unit Disk Graph as follows: the vertex set of the graph is Q and there is an edge between two points in Q iff the distance between them is at most r. For a Unit Disk Graph, we have $r = 2$.

In this paper, we study the following color-spanning set problem: The input is a set S of n points with m colors in the plane. We want to choose m points P from S with m distinct colors such that the $\frac{r}{2}$-Disk Graph on P, $P(r)$, is connected and r is minimized. We call this problem the *minimum connected color-spanning set* problem, abbreviated as **MCCS**. When each color spans at most k points, the problem is denoted as $MCCS(k)$.

While this problem is new, it resembles some of the previous research on "Minimum Spanning Tree with Neighborhoods", etc. Interested readers are referred to [10, 23].

We summarize our results as follows.

1. MCCS(2) is NP-hard to approximate within a factor $3 - \varepsilon$, for some $\varepsilon > 0$.
2. For MCCS(2), we obtain a tight factor-3 approximation.
3. For MCCS(k), we obtain a factor-($2k$-1) approximation.

We discuss the hardness and approximation algorithms for these problems in the following two sections and then conclude the paper in the last section.

2 Hardness of the MCCS Problem

In this section we prove that MCCS is NP-hard even when each color spans at most two points. We prove the NP-hardness of MCCS(2) by a reduction from Planar 3SAT [17], see Figure 2. The Planar 3SAT problem is equivalent to the 3SAT problem restricted to planar formulae.

Theorem 1. *MCCS(2) is NP-hard.*

Proof. We prove the hardness of MCCS(2) by a reduction from Planar 3SAT. Let ϕ be a Boolean formula in conjunctive normal form with n variables $x_1, \ldots, x_n$ in m clauses $\phi_1, \ldots, \phi_m$, each of size at most three. Given the planar embedding of ϕ, we take the following steps to construct a set of points S for of MCCS(2).

For each Boolean variable x_i in ϕ, let k_i^+ and k_i^- be the number of times x_i and $\overline{x}_i$ appears in ϕ respectively, and $k_i = \max\{k_i^+, k_i^-\}$. We use k_i chains labeled with $+$ and k_i chains labeled with $-$. (If $k_i^- < k_i = k_i^+$, then we just make sure $k_i - k_i^-$ chains labeled with $-$ do not connect to any clause; and vice versa. For convenience, we call

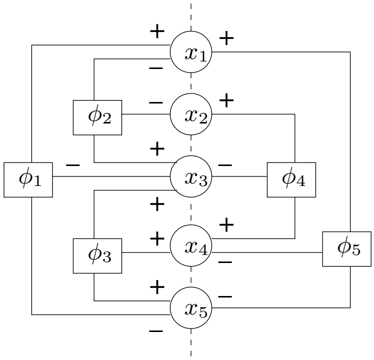

Fig. 2. An instance of planar 3SAT. The circles represent variables, the rectangles represent clauses, the $+$,$-$ on x_i denote the clause connects to literal x_i or $\overline{x}_i$ respectively.

such a chain 'dummy' chain as it does not affect the truth assignment.) These (non-dummy) chains are connected to some fixed points, each of a distinct color (which only appears once and must be selected). These fixed points, denoted by the empty circles in Figure 3, form a variable gadget. Note that the neighboring fixed points have a fixed distance of d_0. See Figure 3.

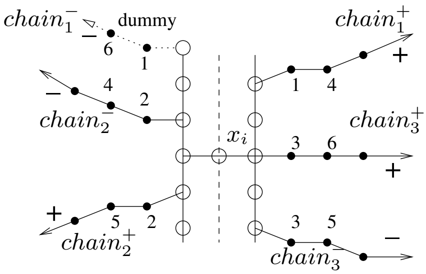

Fig. 3. The variable gadget where each number represents a color

Let the chains labeled with $+$ ($-$) around a variable x_i be sorted in counterclockwise order and let them be $chain_1^+, chain_2^+, ..., chain_{k_i}^+$ and $chain_1^-, chain_2^-, ..., chain_{k_i}^-$ respectively. Then, we use $2k_i$ points with k_i colors (each color spans two points). For each of these k_i colors, we put one point of the j-th color on $chain_i^+$ and the other point of the j-th color on $chain_i^-$. Each of these points is the first point of $2k_i$ chains respectively. This process is repeated until $j = k_i$. In Figure 3, these points correspond to the points labeled with 1, 2 and 3.

We use another $2k_i$ points with k_i colors where each color spans two points. For these k_i colors, we put one point of the j-th color on $chain_i^+$ and the other point of the j-th color on $chain_{i+1}^-$ (we take $chain_1^-$ = $chain_{k_i+1}^-$). This process is repeated until $j = k_i$. Each of these points is the second point of $2k_i$ chains respectively. The distance between the first two adjacent points on any of these $2k_i$ chains is set to be exactly d_0. In Figure 3, these points correspond to the points labeled with 4, 5 and 6.

For the other points on the chains, the sequence is not important. For any color, we just put one point on a chain with label $+$ and the other on a chain with label $-$, as long as there are not two points of the same color on a chain. Starting from the third point on each chain, the adjacent points might have a distance less than or equal to d_0. This will allow us to construct chains of different lengths. Of course, right before a chain reaches a clause, we need to perform something similar to the first two points on each chain. This will be discussed when we cover the clause gadget next.

The idea is that if we need to choose one point for each color to construct the variable x_i, all these points chosen need to be connected (via a communication range of $d_0/2$) to the fixed point of x_i. We either choose the point set on the chains with label $+$, or the point set on the chains with label $-$. The points in different variables have totally different colors. All the fixed points of variables are connected by adding some fixed points, see the dashed line in Figure 3.

For each clause $\phi_p = (x_i \vee x_j \vee \overline{x}_k)$ in ϕ, we add one fixed (clause) point and six points with three colors, two for each color. We try to connect the three chains (corresponding to the three literals in ϕ_p) to the fixed point as follows. We put two points with different colors at the end of each chain such that the three points next to the fixed (clause) point have different colors (e.g., 1,2 and 3 in Figure 4) and they are at distance d_0 to the fixed clause point. Then we connect three chains (in this example, chains with label $+$ for x_i and x_j and with label $-$ for x_k) by using three points in a permutation of these three colors such that the last two points on each chain are of different colors and the distance between them is d_0. See Figure 4. The unique design of the clause gadget makes sure that the fixed point of ϕ_p can only connect to exactly one variable of x_i, x_j, x_k.

Recall that for two intermediate points on a chain, their distance could be less than d_0. For two points p, q from two different chains, we make $d(p, q) > 2d_0$ to ensure that there are no edges between points from different chains.

As the fixed (clause) point for ϕ_p has to connect one of the fixed (variable) point of x_i, x_j, x_k, it is only possible when x_i is true, or x_j is true, or x_k is false. In fact, the clause point ϕ_p can only connect to one variable of x_i, x_j, x_k even if there are more than one literals making ϕ_p true.

Let S be the set of points hence constructed. We finally prove that the planar 3SAT instance ϕ is satisfiable if and only if there is a connected color-spanning $\frac{d_0}{2}$-Disk Graph of S.

"$\rightarrow$": If the planar 3SAT instance ϕ is satisfied, then each clause could connect to one variable. For each variable x_i, we either choose all the points on the chains labeled with $+$, or choose choose all the points on the chains labeled with $-$, which means we choose one point for each color. Let M be the points selected. All the variable points are connected through fixed points, then the $\frac{d_0}{2}$-Disk Graph on M is connected.

"$\leftarrow$": If there is a connected color-spanning $\frac{d_0}{2}$-Disk Graph on a subset of points of S, first notice that in our design of variable and clause gadgets, all the points chosen on the chains between variable x_i and the clause containing x_i or $\overline{x}_i$ must connect to the fixed point of variable x_i. Otherwise, the $\frac{d_0}{2}$-Disk Graph on the chosen points is not connected. According to the configuration of a variable gadget, we either choose the points on $chain_1^+, chain_2^+, ..., chain_{k_i}^+$ or the points on $chain_1^-, chain_2^-, ..., chain_{k_i}^-$.

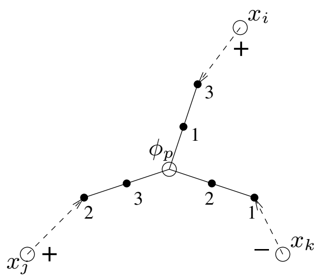

Fig. 4. Clause gadget for $\phi_p = (x_i \vee x_j \vee \overline{x}_k)$. Different numbers mean different colors.

For the $4k_i$ points to construct the first two points for all the variables, we either choose the points on the chains labeled $+$ or the points on the chains labeled $-$. Then we either choose all the immediate points connecting the chains labeled $+$ or the immediate points connecting the chains labeled $-$. (Otherwise, either a clause cannot be connected to any variable it contains, or the subset of points we choose does not span all the colors.) The first case represents the value **True** for this variable, and the second case represents the value **False**. As the fixed point of each clause connects to at least one variable, which means at least one literal in that clause is true, the instance ϕ is hence satisfied.

Therefore, the planar 3SAT instance ϕ is satisfiable if and only if there is a connected color-spanning $\frac{d_0}{2}$-Disk Graph of S. □

With our construction, we have in fact proved that MCCS(2) is NP-hard to approximate with a factor of $2 - \varepsilon$. We can strengthen the result by proving that MCCS(2) is NP-hard to approximate within a factor of $3 - \varepsilon$. We briefly summarize the necessary changes in the next theorem.

Theorem 2. *MCCS(2) is NP-hard to approximate within a factor of* $3 - \varepsilon$*, for some* $\varepsilon > 0$.

Proof. Omitted due to space limitation. □

In the next section, we present approximation algorithms for MCCS(2) and MCCS(k).

3 Approximation Algorithms for MCCS(k)

As a warm-up, we first discuss a special 2SAT instance which will be the basis of our approximation algorithm for MCCS(2). As we will see a bit later, it is a special case of bipartite graphs which always admit a perfect matching following Hall's Marriage Theorem [14]. But the 2SAT formulation is straightforward and is easier for implementation purpose.

Lemma 1. *Let 2SAT(1) be a special instance of 2SAT where a variable* x_i *and its negation* $\overline{x}_i$ *each appears at most once in the instance. Then 2SAT(1) always has a truth assignment.*

Proof. Let ϕ be a 2SAT(1) instance and the clauses are ϕ_1, $\phi_2, \cdots$. Each clause is composed of two literals, e.g., $\phi_j = (x_j \vee y_j)$. $(x_j \vee y_j)$ is equivalent to $\overline{x}_j \rightarrow y_j$ and $\overline{y}_j \rightarrow x_j$. So we have a directed graph $D(\phi)$ on all the literals. If ϕ is not satisfiable, then suppose there is a path $x_i \rightarrow x_2 \rightarrow \cdots x_\ell \rightarrow \overline{x}_i$ in $D(\phi)$ which does not contain nodes x_k and $\overline{x}_k$ between x_i and $\overline{x}_i$ (otherwise, we take a proof for x_k). Then, the clause corresponding to the path is $(\overline{x}_i \vee x_2) \wedge \cdots \wedge (\overline{x}_\ell \vee \overline{x}_i)$. Hence, either the literal $\overline{x}_i$ appears twice in ϕ (a contradiction) or $x_\ell = \overline{x}_2$. If $x_\ell = \overline{x}_2$, then there is a path from x_2 to $\overline{x}_2$ between x_i and $\overline{x}_i$, again a contradiction to the assumption. □

3.1 Approximation Algorithm for MCCS(2)

Given an $\frac{r}{2}$-Disk Graph G with n points and m colors, each node of G is painted with one color, we want to choose a set T of m nodes (one node for each color) from G. We define a graph $H =< T, E' >$, where there is an edge $(u, v) \in E'$ for two nodes $u, v \in T$ if there is a path between u and v of length at most k in G. If H is connected for some value k, we say that H is a $(k-1)$-*hop color-spanning subgraph* of G. In Figure 5, if we select H as node 1 and the remaining doubly labeled nodes from 2 to 6, then H is a 1-hop color-spanning subgraph of G. We now prove the following lemma regarding MCCS(2).

Lemma 2. *Given an $\frac{r}{2}$-Disk Graph G with m colors and each color spans at most two points, if there exists a connected component of G which contains all the m colors, then there is a 2-hop color-spanning subgraph H of G.*

Proof. If this connected component only contains exactly one point z of certain color, then we say z is a *fixed* point. Obviously, a fixed point must be selected to form any color-spanning subgraph. We also do some preprocessing by removing any edge between two nodes of the same color — as such an edge cannot be in any optimal solution. As there exists a connected component of G whose nodes contain all the m colors, we perform a depth-first search on this connected component from a fixed node (point) of G and if there is no fixed point then start with any node. In the searching process, we build a disjoint set of groups, each containing an edge of G, as follows. Let a be the current node which has not been completely explored (see [8]) and let b a neighbor of a in G. If both a and b are not fixed points, and neither a nor b is already in some group, then we build a new group $\{a, b\}$.

Suppose that there are a total of m_1 groups and each group has two points of different colors, hence there are m_2 colors in the m_1 groups with $m_2 \geq m_1$. See Figure 5. In the m_1 groups, if a color paints only one point, then we simply choose that point for H. If a color spans two points in the m_1 groups, we need to choose one for H. We use x_i and $\overline{x}_i$ to denote the two points of color c_i respectively, and each group G_t $(1 \leq t \leq m_1)$ containing two points of color c_i and c_j can be expressed as a clause like $(x_i \vee \overline{x}_j)$). x_i (resp. $\overline{x}_j$) is assigned true when the point of color c_i (resp. c_j) in the group G_t is chosen. Then, the m_1 groups can be expressed as an instance I of 2SAT(1).

By Lemma 1, the above 2SAT(1) instance always has a truth assignment. The truth assignment gives us the selection of the corresponding points for H. If a color spans two points in the connected component but these points never appear in the m_1 groups, then just choose any one of the points for H. Recall that if a color contains only one point in the connected component, we choose that fixed point at the first place for H. Hence we choose m points H from G to have m distinct colors.

Within this connected component of G which contains m distinct colors, from the above construction, it can be seen that between two groups there can be either an edge connecting two nodes from the two groups, or the two groups are connected by a sequence of fixed points. Let these two groups be $G_i = \{a_i, b_i\}$ and $G_j = \{a_j, b_j\}$ respectively. In the worst case, we select one point each from them (say, a_i and b_j) for H, leaving the other two as hops to maintain connectivity in G; i.e., $a_i \to b_i \to a_j \to b_j$. Hence, there are at most three edges (or two hops) in G connecting points in H whose corresponding groups are adjacent.

For any non-fixed point p selected for H which does not belong to any group, p is either adjacent to some fixed point or is adjacent to a point in some group G_t. Otherwise p and one of its neighbors would be forming a new group. Hence, there are at most two edges (or one hop) between p and its nearest point in H.

In summary, if there exists a connected component of G which contains all the m colors, then there is a 2-hop color-spanning subgraph H of G. □

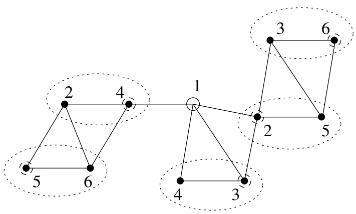

Fig. 5. A connected component of graph G which is divide into groups by DFS and each group has just two points.

The above lemma implies that for nodes in H, if we increase their communication range to $\frac{3r}{2}$, then H will be connected even if all other nodes in $G - H$ are deleted.

Theorem 3. *There is a factor-3 approximation for MCCS(2).*

Proof. It is easy to see that the optimal solution value r^* must be the distance between a pair of points of S which are of different colors. We sort the distances between all pairs of n points of S, and let $d_1, d_2, d_3, ..., d_q$ be the sorted sequence. We try each of d_i as r, for $i = 1, 2, ..., q$, and build the corresponding $\frac{r}{2}$-Disk Graph $G(r)$. Suppose that $G(r)$ contains a connected component which contains of all the m colors the first time when the value of r is increased to d_j, then the optimal solution value r^* satisfies $r^* \geq d_j$. The reason is that if it is not the case, the number of colors of any connected component of $G(r)$, $r < d_j$, is less than m; hence, there are at least two colors whose corresponding points belong to two different components (which is at least d_j distance away). Therefore, when $r < d_j$, it is impossible to find a color-spanning subgraph of $G(r)$ which is connected.

Following Lemma 2, we can compute a 2-hop color-spanning subgraph $H(d_j)$ of $G(d_j)$. In other words, distance between two adjacent points in $H(d_j)$ is at most $3d_j$, as d_j is the maximum length of any edge in a connect component of $G(d_j)$ which contains m colors. This means that we obtain an approximation whose solution value APP satisfies

$$APP \leq 3d_j \leq 3r^*.$$

We finally analyze the time complexity of the algorithm. Computing and sorting $O(n^2)$ distances takes $O(n^2 \log n)$ time. Each time the value of r is increased from r', we either add an edge into a connected component of $G(r')$ (which takes $O(1)$ time) or merge two connected components of $G(r')$ into one (which takes $O(\alpha(n))$ on average — if we use the standard union-find data structure as some auxiliary structure to test whether two elements lie in the same connected component [8,22].) As there are $O(n^2)$ edges and $O(n)$ merges, the total cost is $O(n^2)$. A connected component contains m colors only when there are at least m points in it, that means the graph $G(d_j)$ has at most two connected components satisfying this condition. Hence the time to decide if a connected component contains $m = \Theta(n)$ colors takes $O(n)$ time. When we have a connected component satisfying the condition, the depth-first search, and solving the resulting 2SAT(1) instance takes $O(n)$ time.

Hence the total time complexity is $O(n^2 \log n)$, and the space complexity is $O(n^2)$. □

3.2 Approximation Algorithm for MCCS(k)

For the more general case when each color spans at most k points, we use a similar method as in the previous section until the first time we obtain a $\frac{d_j}{2}$-Disk Graph $G(d_j)$ such that it contains a connected component which contains all the m colors. On any such connected component, we can use the depth-first search (or other method, say a spanning tree) to divide the component into (connected) groups, each containing exactly k points. Then, we choose at least one point from each group (which is proven to be always possible in the next lemma), to form the color-spanning subgraph H. Since in the worst case two points selected are from two neighboring groups, which could be 2(k-1) hops away (or, $2k$-1 edges away), we can give each point selected for H a communication radius of ($2k$-1)d_j to make H connected. Hence, we obtain a factor-($2k$-1) approximation for MCCS(k).

Lemma 3. *In a connected component of $G(d_j)$ which contain all the m colors, if there are g groups of points containing the m colors ($g \leq m$), each group has exactly k points, and each color spans at most k points, then we can always choose one point for each color such that each group has at least one point chosen.*

Proof. We construct a bipartite graph (U, V, E): U denotes the g groups $\{G_1, G_2, ...,$ $G_g\}$, V denotes the m colors $\{c_1, c_2, ..., c_m\}$, and there is an edge between G_i and c_j iff the group G_i contains at least one point of color c_j. Following Hall's Marriage Theorem [14], which, in this setting, states that there is a perfect matching for the bipartite graphs (U, V, E) iff the degree of nodes in U's are at least m, there is a perfect matching where all nodes in U (or groups) will be in a matching. Then, the lemma follows immediately. □

We comment that the 2SAT(1) instance we covered previously is a special case of the bipartite graph we have just discussed. Combined with the previous discussions, it is easily seen that this is gives us a polynomial time approximation. In addition, as the maximum matching algorithm takes $O(n^{5/2})$ time [15] but can be improved to $O(n^2 \log k) = O(n^2)$ time for regular bipartite graphs [7], so the overall running time of this algorithm remains to be $O(n^2 \log n)$. (We comment that with a randomized solution the perfect matching can be computed in $O(n \log n)$ time [13], but it will not change the overall running time of our algorithm.)

Theorem 4. *There is a factor-($2k$-1) approximation for MCCS(k).*

4 Concluding Remarks

We give tight approximation bounds for the Minimum Connected Color-Spanning Set problem, which arises in wireless sensor networks. When k is big, the $2k$-1 factor for MCCS(k) might not be efficient enough. So an interesting question is whether a constant factor approximation can be obtained for MCCS.

References

1. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristan, V.: The farthest color Voronoi diagram and related problems. In: Proceedings of the 17th European Workshop on Computational Geometry (EWCG 2001), pp. 113–116 (2001)
2. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristan, V.: Smallest Color-Spanning Objects. In: Proc. 9th Annu. European Sympos. Algorithms, pp. 278–289 (2001)
3. Beresford, A.R., Stajano, F.: Location privacy in pervasive computing. IEEE Pervasive Computing 2(1), 46–55 (2003)
4. Cheema, M.A., Lin, X., Wang, W., Zhang, W., Pei, J.: Probabilistic Reverse Nearest Neighbor Queries on Uncertain Data. IEEE Trans. Knowl. Data Eng. 22(4), 550–564 (2010)
5. Cheng, R., Kalashnikov, D.V., Prabhakar, S.: Querying imprecise data in moving object environments, knowledge and data engineering. IEEE Transactions on Knowledge and Data Engineering 16(9), 1112–1127 (2004)
6. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving user location privacy in mobile data management infrastrutures. In: Danezis, G., Golle, P. (eds.) PET 2006. LNCS, vol. 4258, pp. 393–412. Springer, Heidelberg (2006)
7. Cole, R., Ost, K., Schirra, S.: On edge coloring bipartite graphs. Combinatorica 21(1), 5–12 (2001)
8. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press (2001)
9. Das, S., Goswani, P.P., Nandy, S.C.: Smallest color-spanning object revised. International Journal of Computational Geometry and Applications 19(5), 457–478 (2009)
10. Efrat, A., Fekete, S.P., Gaddehosur, P.R., Mitchell, J.S.B., Polishchuk, V., Suomela, J.: Improved approximation algorithms for relay placement. In: Halperin, D., Mehlhorn, K. (eds.) ESA 2008. LNCS, vol. 5193, pp. 356–367. Springer, Heidelberg (2008)
11. Fleischer, R., Xu, X.: Computing Minimum Diameter Color-Spanning Sets. In: Lee, D.-T., Chen, D.Z., Ying, S. (eds.) FAW 2010. LNCS, vol. 6213, pp. 285–292. Springer, Heidelberg (2010)

12. Gedik, B., Liu, L.: A customizable k-anonymity model for protecting location privacy. In: Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS 2005), pp. 620–629 (2005)
13. Goel, A., Kapralov, M., Khanna, S.: Perfect matchings in $O(n \log n)$ time in regular bipartite graphs. In: Proceedings of STOC 2010, pp. 39–46 (2010)
14. Hall, P.: On representatives of subsets. J. London Math. Soc. 10(1), 26–30 (1935)
15. Hopcroft, J., Karp, R.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comput. 2(4), 225–231 (1973)
16. Ju, W., Fan, C., Luo, J., Zhu, B., Daescu, O.: On Some Geometric Problems of Color-Spanning Sets. J. of Combinatorial Optimization 26(2), 266–283 (2013)
17. Lichtenstein, D.: Planar formulae and their uses. SIAM J. Comput. 11(2), 329–343 (1982)
18. Mumey, B., Spendlove, K., Zhu, B.: Extending the lifetime of a WSN by partial covers. In: Proceedings of IEEE ICC 2013 (2013)
19. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic Skylines on Uncertain Data. In: Proc. of VLDB 2007, pp. 15–26 (2007)
20. Pfoser, D., Jensen, C.S.: Capturing the uncertainty of moving-objects representations. In: Güting, R.H., Papadias, D., Lochovsky, F.H. (eds.) SSD 1999. LNCS, vol. 1651, pp. 111–131. Springer, Heidelberg (1999)
21. Sistla, P.A., Wolfson, O., Chamberlain, S., Dao, S.: Querying the uncertain position of moving objects. In: Etzion, O., Jajodia, S., Sripada, S. (eds.) Temporal Databases-Research and Practice. LNCS, vol. 1399, pp. 310–337. Springer, Heidelberg (1998)
22. Tarjan, R.: Efficiency of a good but not linear set union algorithm. J. of ACM 22(2), 215–225 (1975)
23. Yang, Y., Lin, M., Xu, J., Xie, Y.: Minimum spanning tree with neighborhoods. In: Kao, M.-Y., Li, X.-Y. (eds.) AAIM 2007. LNCS, vol. 4508, pp. 306–316. Springer, Heidelberg (2007)
24. Yuen, S.-M., Tao, Y., Xiao, X., Pei, J., Zhang, D.: Superseding Nearest Neighbor Search on Uncertain Spatial Databases. IEEE Trans. Knowl. Data Eng. 22(7), 1041–1055 (2010)
25. Zhang, D., Chee, Y.M., Mondal, A., Tung, A.K.H., Kitsuregawa, M.: Keyword search in spatial databases: Towards searching by document. In: Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE 2009), pp. 688–699 (2009)
26. Zhang, H., Hou, J.C.: Maintaining sensing coverage and connectivity in large sensor networks. Ad Hoc & Sensor Wireless Networks, an Intl J. 1, 89–124 (2005)
27. Zhang, H., Nixon, P., Dobson, S.: Partial coverage in homological sensor networks. In: Proceedings of the 5th IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob 2009), pp. 42–47 (2009)